64. A central processing unit comprising at least a register file, an execution unit, the central processing unit adapted to decode a plurality of instruction sets; one of the instruction sets being stack-based instructions; at least some of the operands for the plurality of instruction sets maintained in said register file; the central processing unit, including a unit adapted to produce at least one of overflow or underflow for the operands for stack-based instructions maintained in the register file.

65. A system comprising:

a first unit adapted to execute register-based instructions, the first unit having an associated register file; and

a hardware unit associated with the first unit, the hardware unit adapted to decode stack-based instructions and store portions of the operand stack in the first unit's register file, wherein the hardware unit is adapted to receive at least some stack-based instructions and register-based instructions from a cache.

66. A central processing unit having a register file, an execution unit for executing register-based instructions, the central processing unit adapted to decode stack-based instructions, wherein a portion of the operand stack associated with the stack-based instructions is stored in the said register file and at least one of overflow or underflow indication is produced for the portion of the operand stack stored in said register file; the

central processing unit receiving at least some of the register-based instructions and stack-based instructions from a cache.

67. The central processing unit of claim 66 wherein the register-based instructions and stack-based instructions are from the same cache

68. The central processing unit of claim 66 wherein the cache is an instruction cache

69. A central processing unit having at least a register file and a execution unit adapted to execute register-based instructions, a unit adapted to execute stack-based instructions using the said execution unit, wherein a portion of the operand stack associated with the stack-based instructions is stored in the register file wherein an indication of at least one of overflow or underflow is produced and wherein some of the stack-based instructions produce references to at least two of the registers in the said register file.

70. The central processing unit of claim 69 wherein the two registers referenced are the top two stack elements of the operand stack.

71. A central processing unit comprising at least a register file, an execution unit, the central processing unit adapted to execute register-based instructions and stack-based instructions, the central processing unit having a common program counter register for the

register-based instructions and stack-based instructions wherein at least some of the operands for the plurality of instruction sets is maintained in said register file; the central processing unit, including a unit adapted to produce at least one of overflow or underflow for the operands for stack-based instructions maintained in the register file.

72. A central processing unit comprising at least a register file, an execution unit, the central processing unit adapted to execute register-based instructions and stack-based instructions using said register file and execution unit, the central processing unit having an indication as to which instruction set is being executed, the central processing unit, including a unit adapted to produce at least one of overflow or underflow for the portion of the operand stack for stack-based instructions maintained in the register file.

73. A central processing unit comprising at least a register file, an execution unit, the central processing unit adapted to execute register-based instructions, the central processing unit including a unit to decode and execute at least some stack-based instructions using said register file and execution unit, some of the stack-based instructions causing an exception wherein they are executed in software, the central processing unit, including a unit adapted to produce at least one of overflow or underflow for the portion of the operand stack for stack-based instructions maintained in the register file.

74. The central processing unit of claim 73 wherein the some of the stack-based instructions, which are executed in software are done by issuing said register-based instructions.

75. The central processing unit of claim 73 wherein at least some of the plurality of instruction sets instructions are from the same instruction cache.

76. A central processing having a first and a second register file, an execution unit, the central processing unit adapted to execute register-based instructions using at least the execution unit and the first register file, the central processing unit adapted to decode stack-based instructions, execute at least some of the stack-based instructions using the execution unit wherein at least a portion of the operand stack is stored in the second register file, wherein the central processing unit is adapted to produce an indication of at least one of overflow or underflow for the portion of the operand stack stored in the second register file.

77. The central processing unit of claim 76 wherein the Java registers are stored in the first register file when running a virtual machine.

78. The central processing unit of claim 76 wherein the variables are stored in the first register file.

79. The central processing unit of claim 76 wherein a store or a load instruction is executed due to an overflow or underflow indication respectively.

80. A central processing unit having at least a register file and a execution unit, the central processing unit adapted to execute register-based instructions and adapted to receive stack-based instructions, decode the stack-based instructions and execute the stack based instructions using the said execution unit and register file, wherein at least a portion of the operand stack for the stack-based instructions is stored in the said register file of the central processing unit; the central processing unit producing an indication of at least an overflow or underflow for the portion of the operand stack in the register file and having a register for OpTop which is at least incremented or decremented.

81. A central processing unit comprising at least a register file, an execution unit, the central processing unit adapted to decode a plurality of instruction sets; one of the instruction sets being stack-based instructions; at least some of the operands for the plurality of instruction sets maintained in said register file; the central processing unit executing at least some of the stack-based instructions which are branch instructions utilizing the register file and execution unit, the central processing unit, including a unit adapted to produce at least one of overflow or underflow for a portion of the operand stack for stack-based instructions maintained in the register file.

82. The central processing unit of claim 81 wherein the branch instructions for java byte codes includes at least one of ifeq, ifne, iflt, ifge, ifgt, ifle, if_icmpeq, if_icmpne, if_icmplt, if_acmpge, if_cmpgt, if_icmple, if_acmpeq, if_acmpne, ifnull, ifnonull, lcmp, fcmpl, fcmpg, dcmpl, or dcmpg.

83. The central processing unit of claim 81 wherein the execute logic produces a branch indication for at least some of the branch instructions.

84. A system comprising;

a first unit adapted to execute register-based instructions;

a hardware unit associated with the first unit, the hardware unit adapted to decode stack-based instructions, store operands for stack based instructions in the first units register file wherein an underflow or overflow indication is produced for the portion of the operand stack stored in the register file; the hardware decoding at least some of the branch instructions and causing execution of the branched instructions by the first unit's execution unit.

85. The system of claim 84 wherein the branch instructions for java byte codes includes at least one of ifeq, ifne, iflt, ifge, ifgt, ifle, if_icmpeq, if_icmpne, if_icmplt, if_acmpge, if_cmpgt, if_icmple, if_acmpeq, if_acmpne, ifnull, ifnonull, lcmp, fcmpl, fcmpg, dcmpl, or dcmpg.

86. The system of claim 84 wherein the execute logic of the first unit produces a branch indication for at least some of the branch instructions

87. A central processing unit comprising at least a register file, an execution unit, the central processing unit adapted to decode a plurality of instruction sets, at least one of the instruction sets being stack-based instructions; an instruction buffer wherein at least one of register-based instructions or stack-based instructions are loaded, a stack-based instruction decoder producing an indication of the number of bytes associated with the stack-based instruction being decoded wherein said number of bytes are removed from the instruction buffer, the plurality of instruction sets being executed by the said execution unit.

88. The central processing unit of claim 87 wherein at least a portion of the operand stack associated with the stack-based instructions is stored in the said register file and wherein the central processing unit produces an indication of at least one of overflow or underflow for the portion of the said operand stack.

89. The central processing unit of claim 87 wherein a branch taken indication flushes stack-based instructions from the instruction buffer.

90. A central processing unit comprising at least a register file, an execution unit, the central processing unit adapted to decode a plurality of instruction sets; one of the

instruction sets being stack-based instructions; at least some of the operands for the plurality of instruction sets maintained in said register file producing an indication of at least one of overflow or underflow for a portion of the operand stack associated with the stack-based instructions maintained in the register file, the central processing unit adapted to store operand data from some of the stack-based instructions in the said register file.

91. The central processing unit of claim 90 wherein the stack-based instruction is a SiPush instruction.

92. The central processing unit of claim 90 wherein the stack-based instruction is a BiPush instruction.

93. The central processing unit of claim 91 and claim 92 wherein the some of the operand data is from an instruction buffer.

94. The central processing unit of claim 90 wherein some of the operand data from the some of the stack-based instructions is sign extended.

95. A central processing unit comprising at least a register file, an execution unit, the central processing unit adapted to decode a plurality of instruction sets; at least one of the instruction sets being register-based and at least another instruction set being stack-based

instructions and at least some of the operands for the plurality of instruction sets maintained in said register file; the central processing unit decoding and executing register-based instruction upon powering up and when a virtual machine is invoked, changing modes to decode and execute stack-based instructions, using the said register file and execution unit for plurality of instruction sets and wherein at least one of overflow or underflow indication is produced for the portion of the operand stack for the stack-based instructions stored in the said register file.

96. The central processing unit of claim 95 wherein a second register file is operably connected to the execution unit.

97. The central processing unit of claim 96 wherein at least a portion of the operand stack for the stack based instructions is stored in at least one register file.

98. A central processing unit adapted to execute register-based instructions, having at least a register file and an execution unit; a stack-based language accelerator integrated within the said central processing unit wherein the stack-based accelerator produces the native instructions operated on by the main portion of the central processing unit and wherein at least some of the register-based instructions and at least some of the stack-based instructions reside in the same cache.

99. The central processing unit of claim 98 wherein the execution logic of the central processing unit executes the native instructions.

100. The central processing unit of claim 98 wherein the cache is an instruction cache.

101. A system comprising a first unit adapted to have at least an execution unit, a hardware unit associated with the first unit, the hardware unit adapted to decode stack-based instructions and execute said instructions using the execution unit wherein at least some of the stack based instructions executed are branch instructions and at least some of the stack-based instructions cause an exception whereby they are executed in software.

102. The system of claim 101 wherein at least some of the Java registers are maintained in hardware.

103. The system of claim 101 wherein at least some of the stack-based instructions are from a cache

104. The system of claim 101 wherein the branch instructions for java byte codes includes at least one of ifeq, ifne, iflt, ifge, ifgt, ifle, if_icmpeq, if_icmpne, if_icmplt, if_acmpge, if_cmpgt, if_icmple, if_acmpeq, if_acmpne, ifnull, ifnonull, lcmp, fcmpl, fcmpg, dcmpl, or dcmpg.

105. The system of claim 101 wherein the execute logic produces a branch indication for at least some of the branch instructions.

106. The central processing unit of claims 64, 66, 69, 71, 72,73,76,80,81,83, 84, 86, 95 and 98 where the stack-based instructions are Java byte codes.

107. The system of claims 63, 65, and 101 where the stack-based instructions are Java byte codes.

108. The system of claims 63, 65, and 101 where the hardware unit is part of the central processing unit.